

LEADER

北京同立钧成知识产权代理有限公司
Leader Patent & Trademark Firm

A-902, TOPBOX Building, No.69 Beichen West
Road, Chaoyang District, Beijing 100029, P.R. China
Phone: 86-10-58773108/9
Fax: 86-10-58773700
Website: <http://www.infopatent.com.cn>
E-mail: international@infopatent.com.cn

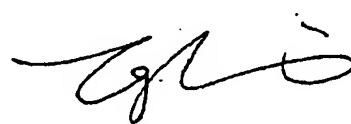
May 4, 2008

STATEMENT

To whom it is concerned,

This is to certify that the annexed is a true and accurate
copy of the English translation of Chinese priority application
No. 200410008437.6.

Sincerely yours,



Mr. ZANG Jianming, Senior Partner

Patent Attorney

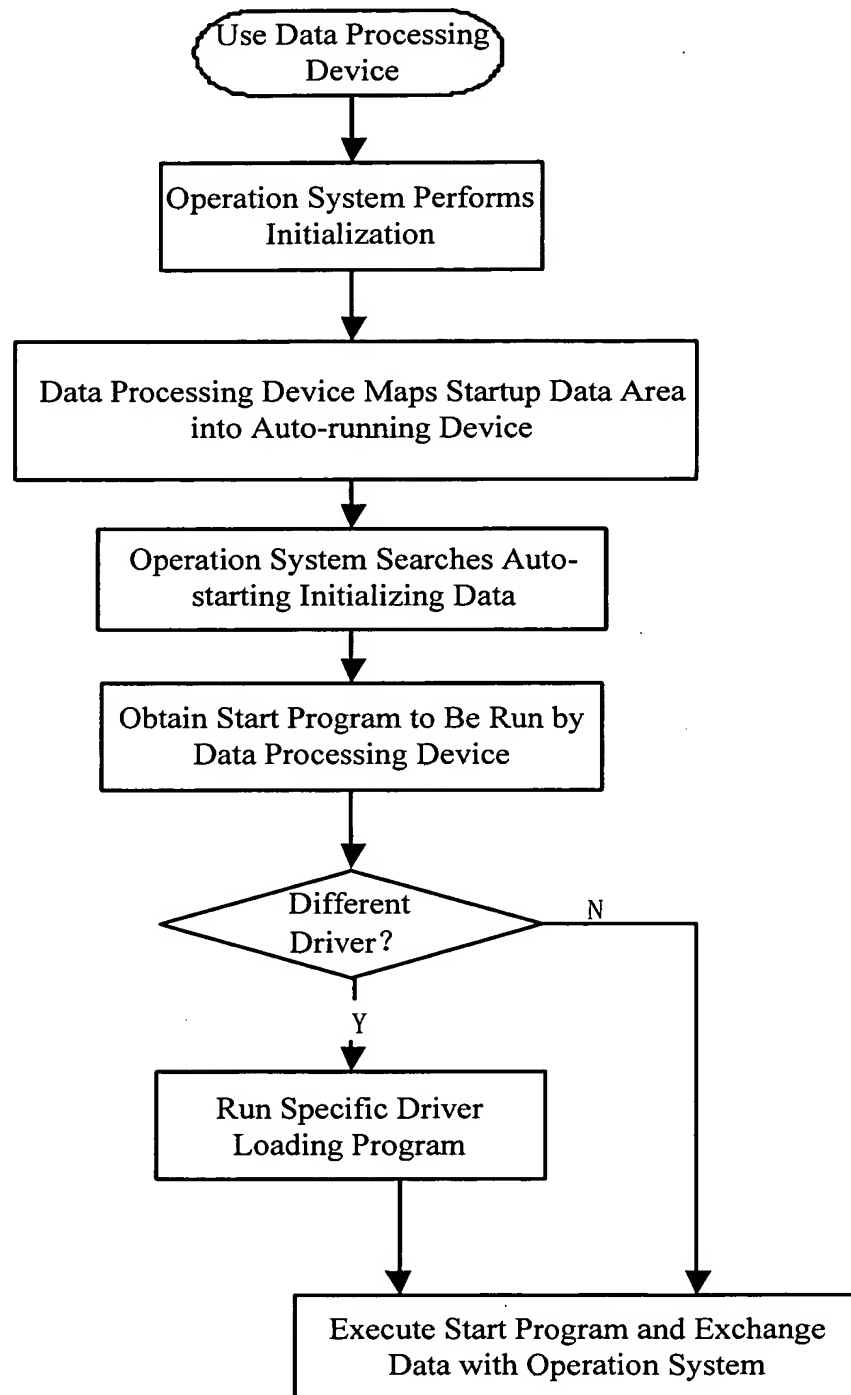


Ms. MENG Jinzhe,

Assistant Patent Attorney

Abstract

The invention relates to a method for data processing device with a standard data interface exchanging data with the operating system. The method includes: the initial data for auto-running, the start program and so on are written into the startup data area; when the data processing device is used, the operation system performs necessary initialization of the data processing device in accordance with the standard data interface; the data processing device maps the startup data area into the device with auto-running attribute by default of the operation system; the operation system searches the startup data area for the initial data for auto-running, and from which acquires the start program to be run by the data processing device and the running order; if the driver needed by the data processing device differs with the standard driver embedded in the operation system, the specific driver loading program of the startup data area is run; execute the start program in running order and exchange the data with the operation system. By using the invention, it can realize the data processing device automatically exchanging data with the operating system, and the exchanged data can be defined by user.

Abstract Drawing

What is claimed is:

1. A method for data processing device exchanging data with operating system, in which said data processing device includes a standard data interface, a control module and a storage module; the standard data interface used for transmitting data between the data processing device and the operating system; the control module used for controlling the storage module to achieve data exchanging with the operation system; the storage module having a startup data areas for presetting initial data for auto-running and data needed for exchanging data; and the method includes the following steps:

step 1, when the data processing device is used, the operation system performs necessary initialization of the data processing device in accordance with the type of data interface;

step 2, the data processing device maps the startup data area into the device with auto-running attribute by default of the operation system;

step 3, the operation system searches the startup data area under the pattern of auto-running for the auto-running initializing data, and acquires the start program to be run by the data processing device and running order;

step 4, if the driver needed by the data processing device differs with the standard driver embedded in the operation system, the specific driver loading program of the startup data area is run;

step 5, execute the start program in running order and exchange the data with the operation system.

2. The method for data processing device exchanging data with operating system as claimed in claim 1, wherein if one or a plurality of data storage areas are setup in said storage module for storing configuration information and /or data to be exchanged; and said

step 5 further comprises: execute the start program in the running order; the data processing device maps each data storage area into other type of devices; and the start program parses the configuration information in each data storage area and exchanges the data with the operation system in accordance with the configuration information.

3. The method for data processing device exchanging data with operating system as claimed in claim 1 or 2, wherein if the storage module comprises a plurality of startup data areas, said step 2 and step 3 comprises: the data processing device maps the multiple startup data areas into devices with auto-running attribute by default of multiple operation systems in the same time; the operation systems search the startup data areas by turns under the pattern of auto-running for the auto-running initial data and obtain the startup program to be run by the data processing device and running order.

4. The method for data processing device exchanging data with operating system as claimed in claim 1, wherein the standard interface of said data processing device is USB interface or IEEE1394 interface; and the device with auto-running attribute by default of the operation system is CD driver.

5. The method for data processing device exchanging data with operating system as claimed in claim 4, wherein if said data processing device is a data processing device with the USB interface, the method for exchanging data comprises:

perform initialization of the data processing device, and write the initial data for auto-running, the start program and data required for achieving data exchanging into the startup data area of the data processing device;

when the data processing device is used, the operation system performs necessary initialization of USB interface through the

embedded driver program of USB interface;

the data processing device maps the startup data area into the CDROM device;

the operation system searches the startup data area under the CDROM
5 pattern for the initializing file for auto-running, and from which
acquires the start program to be run by the data processing device
and the running order;

if the driver needed by the data processing device differs with the
standard driver embedded in the operation system, the specific driver

10 loading program of the startup data area is run;

execute the start program in the running order;

the data processing device maps each data storage area into other
type of devices; and

the start program parses the configuration information in each data
15 storage area and exchanges the data with the operation system in
accordance with the configuration information.

METHOD FOR DATA PROCESSING DEVICE EXCHANGING DATA WITH OPERATING SYSTEM

Field of the Invention

The invention relates to a method for data processing device exchanging data with the operating system, in particular, to a method for data processing device with standard data interface exchanging data with the operating system.

Background of the Invention

In general cases, after connecting with a computer, a data processing device will exchange data with the operating system. There are different processes of data exchanging depending on whether the data processing device is plug_and_play or not.

If the data processing device is plug_and_play, the process of data exchanging is as following: The operation system auto-checks whether there is plug_and_play device during the system start, if a new device is found out and there's *.inf file of such kind device in the direction INF of the operation system, then the operating system auto-installs driver program, else the operating system will start the hardware guide to prompt user to select or search driver program of the device, i.e. the corresponding *.inf file, then the hardware guide will copy the specified file to corresponding direction according to the content of the *.inf file, and write corresponding information to register to finish installing of the driver program of the device. After finishing installation, setting the attribute of the device is necessary. For example, before using a network card, the network protocol must be installed and set; before using modem to access network, "new connection" must be built firstly, and so on. And then the operation system starts to exchange data with the device. The user manually starts corresponding application program or perform operation of the device and exchange data with

the operating system to realize the fixed function.

If the data processing device is not plug_and_play, the process of data exchanging is as following: The operation system can not auto-check the device during the system start and the user need to start the hardware guide for installation directly and manually. If there's not the corresponding *.inf file in the direction INF, the operating system will start the hardware guide to prompt user to select or search driver program of the device, i.e. the *.inf file. The hardware guide will copy the specified file to corresponding direction according to the content of the *.inf file and write corresponding information to register to finish installing of the driver program of the device. After finishing installation, the attribute of the device is set. For example, before using a network card, the network protocol must be installed and set; before using modem to access network, "new connection" must be built firstly, and so on. And then the operating system starts to exchange data with the device. The user manually starts corresponding application program or perform operation of the device and exchange data with the operating system to realize the fixed function.

In the prior art, some problems are existed in the method for data processing device exchanging data with the operating system: 1) If the driver program of some kind data processing device have not be embedded in the operation system, then the user must manually install the driver program of the device and the process of the operation is complicated. 2) Although the standard driver program of some kind data processing device has been installed in the operation system, but if the user need to use the substandard driver program of the device, then the user also must manually install the substandard driver program. 3) After finishing installing the driver program of data processing device, the user also demands to set relational device

parameters of some special devices to make the normal use of the devices. This also causes the complicated process of operation. 4) When using the data processing device to attain definite application function, the user must start relative application manually. This also causes the discommodious operation.

Summary of the Invention

The technical problem to be solved by the present invention is to provide a method for data processing device with standard data interface exchanging data with the operating system. The data processing device may automatically exchange definable data with the operating system to attain the fixed functions without user's interference.

A method for data processing device exchanging data with operating system, said data processing device including a standard data interface, a control module and a storage module; the standard data interface used for transmitting data between the data processing device and the operating system; the control module used for controlling the storage module to achieve data exchanging with the operation system; the storage module having a startup data areas for presetting initial data for auto-running and data needed for exchanging data; and the method includes the following steps:

step 1, when the data processing device is used, the operation system performs necessary initialization of the data processing device in accordance with the type of data interface;

step 2, the data processing device maps the startup data area into the device with auto-running attribute by default of the operation system;

step 3, the operation system searches the startup data area under the pattern of auto-running for the auto-running initializing data ,

and acquires the start program to be run by the data processing device and running order needed;

step 4, if the driver needed by the data processing device differs with the standard driver embedded in the operation system, the specific driver loading program of the startup data area is run;

step 5, execute the start program in running order and exchange the data with the operation system.

If one or a plurality of data storage areas are setup in said storage module for storing configuration information and /or data to be exchanged; and said step 5 further comprises: execute the start program in the running order; the data processing device maps each data storage area into other type of devices; and the start program parses the configuration information in each data storage area and exchanges the data with the operation system in accordance with the configuration information.

If the storage module comprises a plurality of startup data areas, said step 2 and step 3 includes: the data processing device maps the multiple startup data areas into devices with auto-running attribute by default of multiple operation systems in the same time; the operation systems search the startup data areas by turns under the pattern of auto-running for the auto-running initial data and obtain the startup program to be run by the data processing device and running order.

Using the present invention can achieve the data processing device automatically exchanging data with the operation system, and the exchanged data can be defined by the user. Therefore, the deficiency of the prior art, complicated operation, can be solved.

Brief Description of the Drawings

FIG.1 is an inside schematic view of the data processing device

with standard interface adapted to the method of the present invention;

FIG.2 is a flow chart of a method of the data processing device shown in FIG.1 exchanging data with the operating system;

FIG.3 is a flow chart of exchanging data when the data processing device is a MP3 player appending USB interface;

FIG.4 is a flow chart of exchanging data when the data processing device is a Flash card reader appending USB interface;

FIG.5 is a flow chart of automatically update function of a MP3 player produced by the method of the present invention.

Detailed Description of the Preferred Embodiments

The present invention will be described in details with reference to the accompanying drawings and the preferred embodiments.

As shown in FIG.1, the data processing device 11 includes: a standard data interface 13, a control module 14 and a storage module 15, wherein, the standard data interface 13 is performed as an interface for transmitting data between the data processing device 11 and the computer 12, the control module 14 is used for controlling the storage module to achieve data exchanging with the operation system, the storage module 15 is used for storing data.

The standard data interface 13 can be universal serial bus(USB) interface or IEEE1394 interface or CF interface and so on. The control module 14 divides the storage module 15 into N data areas, and maps one of data areas into the device which is capable of auto-running by default of the operation system, furthermore maps the remained N-1 data areas into a plurality of devices of other types.

The storage module 15 includes one or multiple startup data areas 16, for storing the data used for defining the regulation of data exchanging, such as initial data for auto-running and start program

etc and common data to be exchanged. In addition, the storage module 15 further includes one or multiple data storage areas 17, in which the data content pointed by data exchanging is stored. The data content may include only the configuration information or the data to be exchanged, or include the configuration information and the data to be exchanged together. Wherein, the configuration information is used to define the position, exchanging order and exchanging pattern of the data to be exchanged in the data storage areas.

In the present invention, the startup data area 16 can be setup as visible area or invisible area to user, or setup as read-only area or R/w area, furthermore setup as the area protected by the security mechanism. The storage data area 17 can be setup as visible area or invisible area to user, or setup as read-only area or R/w area, furthermore setup as the area protected by the security mechanism.

The storage module 15 of the data processing device 11 can use Nand Flash, Nor Flash, electrically erasable programmable read only memory EEPROM and hard disk, etc.

The data processing device incorporates different electronic products, such as USB mobile storage, MP3 player, reader, personal digital assistant (PDA), databank, electronic book(E-book) and electronic phone(E-phone), digital camera and recorder pen, etc.

Through the method of the present invention for data processing, during the practical application the above-presented data processing device performs automatically exchanging definable data with the operation system without user's interference. **Refer to FIG.2, it's a flow chart of the data exchanging of the present invention.**

The data which is required for starting auto-running initializing data in the startup data area and perform data exchanging can be preset. Write the data which is used to define the regulations of data

exchanging, including auto-running initializing data, start program, and the data required for implementing data exchanging, into the startup data area of the data processing device. The data required for implementing data exchanging includes the data needed to be exchanged and/or specific driver loading program and so on.

When the data processing device is plug into the computer, the operation system performs necessary initialization of the data processing device in accordance with the type of data interface of the data processing device. The data processing device maps the startup data area into the device with auto-running attribute by default of the operation system, and the operation system searches the startup data area under the pattern of auto-running for the auto-running initializing data to get the start program and running order needed to be run by the data processing device. If the driver needed by the data processing device differs with the standard driver embedded in the operation system, then the specific driver loading program of the startup data area is run. And then the data processing device executes the start program in running order to call the data in the data area and exchange the data with the operation system.

The data storage area is setup in the storage module of the data processing device to store data to be exchanged. In the case, after the start program being executed, the data processing device maps the data storage area into other devices, and then the start program calls the data in the data storage area and exchanges the data with the operation system.

The data storage area of the storage module also may store configuration information which defines the position, exchanging order and exchanging pattern of the data to be exchanged in the data storage area. Here, the data to be exchanged can be stored not only in the startup data area but the data storage area. Under this

condition, after the start program is executed, the data processing device maps the data storage area into other type of device such as USB magnanimity storage device etc. Then the start program parses the configuration information in the data storage area and calls the data in accordance with configuration information and exchanges the data with the operation system.

If there's a plurality of startup data areas in the storage module of the data processing device, then the data processing device will map the multiple startup data areas into devices with auto-running attribute by default of multiple operation systems in the same time. Then the operation systems search the startup data areas by turns under the pattern of auto-running for the auto-running initial data and obtain the startup program to be run by the data processing device and running order.

At present, the standard interface include USB interface or IEEE1394 interface, while the devices with auto-running attribute by default of the operation system mainly include CD driver (CDROM) and hard disk etc. Three exemplary embodiments will be provided to describe the application of the present invention.

Exemplary embodiment 1

The data processing device is a MP3 player with an USB interface which uses Flash as the storage medium and has one startup data area and one data storage area. Wherein the startup data area is a read-write storage area visible to user, so does the data storage area which inherits the intrinsic fundament functions of storage medium. When the MP3 player is connected to the computer, the medium files in storage can be auto-played.

Before practically using the MP3 player, there's supposed to complete ahead: writing the medium files to be played into the data storage area, compiling auto-running initializing file Autorun.ini,

setting the start program of auto-play medium files and driver program of the MP3 player, and writing the startup data including modified auto-running initializing file Autorun.ini, the start program of auto-play medium files and driver program of the MP3 player etc into the startup data area.

As shown in FIG.3, when the MP3 player is plugged into the computer, the operation system performs necessary initialization of USB interface through the embedded driver program of USB interface. After finishing initialization, The MP3 player maps the startup data area into the CDROM device capable of auto-running and loads the startup data area of the MP3 player, and the operation system searches the startup data area under the CDROM pattern for the initializing file Autorun.ini and from which acquires the start program to be run by the MP3 player and the running order. The driver program of the MP3 player differs with the driver program by default of the operation system, thus the driver program in the startup data area is run, and the driver program by default of the operation system is update with the driver program of the MP3 player. While the start program is executed, the data storage area of the MP3 player is loaded and mapped into other types of device. Then the medium files stored in the data storage area are run according to the configuration information and the MP3 player is in the pattern of monitoring until all the operations are stop or the MP3 player is pulled out.

The above-said startup data area and data storage area of the MP3 player are both set as read-write storage area visible to user, thus, user can define the medium files to be played and other matters.

Exemplary embodiment 2

The data processing device is a Flash card reader with an USB interface. When a Flash card is plugged into the reader, the data stored in the Flash card can be accessed, and then the reader

automatically performs data-predefined exchanging with the operation system according to different cards.

Assuming that there are Flash card A and Flash card B, wherein Flash card A has one startup data area which is setup as read-only storage area invisible to user and hasn't data storage area, while Flash card B has One startup data area which is setup as read-only storage area invisible to user and two data storage areas, one of which is read-write storage area invisible to user and the other is read-write storage area visible to user.

The function implemented by the Flash card A is to print the content in the card by printer connected with the computer. While the function implemented by the Flash card B is to synchronize the data stored in secure data storage area of the card with the computer and play designated medium files in the common data storage area.

Before practically using, there's supposed to perform initialization of Flash card A and Flash card B respectively. The initialization of the Flash card A comprises: compiling auto-running initializing file Autorun.ini, setting the start program to be executed, i.e. transmitting the present content of the Flash card A to the computer by infrared interface, printing with installed infrared transmission driver program and printer driver program, and then writing the data including modified initializing file Autorun.ini, infrared transmission driver program, printer driver program and start program into the startup data area. The initialization of the Flash card B comprises: writing the medium files to be auto-played into the common data storage area, compiling the auto-running initializing file Autorun.ini, setting the start program to be executed i.e. synchronizing the data stored in the secure data storage area of Flash card B with the computer and auto-playing the start program of medium file, and then writing the

data including modified initializing file Autorun.ini etc into the startup data area.

Please refer to FIG.4, when the reader plugged with Flash card A is plugged into the computer, the operation system performs the necessary initialization of USB interface though the embedded driver program of USB interface, and then the reader plugged with Flash card A maps the startup data area of the Flash card A into CDROM device capable of auto-running and loads the startup data area of the Flash card A. The operation system searches the startup data area of Flash card A under the CDROM pattern for the initializing file Autorun.ini, from which acquires the start program to be run by the reader, loads and executes the infrared transmission driver program and printer driver program stored in the startup data area to make the present content of the Flash card A transmitted to the computer by the infrared transmission and printed. And then the reader is in the pattern of monitoring until all the operations are stop or the reader is pulled out.

When the reader plugged with Flash card B is plugged into the computer, the operation system performs the necessary initialization of USB interface though the embedded USB interface driver program. After the initialization finishes, the reader maps the startup data area of the Flash card B into CDROM device capable of auto-running and loads the startup data area of the Flash card B. The operation system searches the startup data area of Flash card B under the CDROM pattern for the initializing file Autorun.ini, from which acquires the start programs to be run by the reader and the running order, executes the start programs by turns, loads and maps the data storage area of the Flash card B into other types of device. And the application program is run to synchronize data with the computer and the medium file in the common data storage area. Then the reader is in the pattern

of monitoring until all the operations are stop or the reader is pulled out.

Exemplary embodiment 3

To realize auto-update. The data processing device is a MP3 player with an USB interface which uses Flash as the storage medium and has one startup data area and one data storage area. Wherein the startup data area is a read-write storage area visible to user, so does the data storage area which inherits the intrinsic fundament functions of storage medium.

When the MP3 player is connected to the computer, the medium files in the MP3 player can be auto-played. At one time, if the computer connects with the internet, the medium files stored in the MP3 player will be auto-updated. Thus, when the MP3 player is connected with the computer again, the updated medium files will be auto-played.

To set the relational data of the MP3 player comprising: compiling the auto-running initializing file Autorun.ini, setting the start program of the auto-playing medium files and the driver program of the MP3 player, writing the data including modified initializing file Autorun.ini, the driver program of the MP3 player and the start program etc into the data startup area, writing the medium files to be auto-played into the data storage area and writing the auto-updated application program into the data storage area.

AS shown in FIG.5, when the MP3 player is plugged into the computer, the operation system performs necessary initialization of USB interface though the embedded driver program of USB interface. After finishing initialization, the MP3 player maps the startup data area into the CDRom device capable of auto-running and loads the startup data area, and the operation system searches the startup data area under the CDRom pattern for the initializing file Autorun.ini and from which acquires the start program to be run by the MP3 player

and the running order. The MP3 player needs specific driver program, thus the driver program of the MP3 player stored in the startup data area is run, and the driver program by default of the operation system is updated with the specific driver program of the MP3 player. The start programs are executed by turns and the data storage area of the MP3 player is loaded and mapped into other types of device. Then the medium files stored in the data storage area are run, the start program parses the configuration information stored in the data storage area and the auto-update program is run to update the medium files stored in the data storage area. The MP3 player is in the pattern of monitoring until all the operations are stop or the MP3 player is pulled out.

When the MP3 player is plugged into the computer again, the operation system performs necessary initialization of USB interface through the embedded driver program of USB interface. After finishing initialization, the MpP3 player maps the startup data area into the CDROM device capable of auto-running and loads the startup data area, and the operation system searches the startup data area under the CDROM pattern for the initializing file Autorun.ini and from which acquires the start program to be run by the MP3 player and the running order. The MP3 player needs specific driver program, thus the driver program of the MP3 player stored in the startup data area is run, and the driver program by default of the operation system is updated with the specific driver program of the MP3 player. The start programs are executed by turns and the data storage area of the MP3 player is loaded and mapped into other types of device. The updated medium files stored in the data storage area are run. Based on the same principle, the present invention also can be used to realize the definable auto-logon, auto-registration application, and so on.

It should be understood that the above embodiments are used only

to explain, but not to limit the present invention. In despite of the detailed description of the present invention with referring to above preferred embodiments, it should be understood that various modifications, changes or equivalent replacements can be made by those skilled in the art without departing from the spirit and scope of the present invention and covered in the claims of the present invention.

Drawings

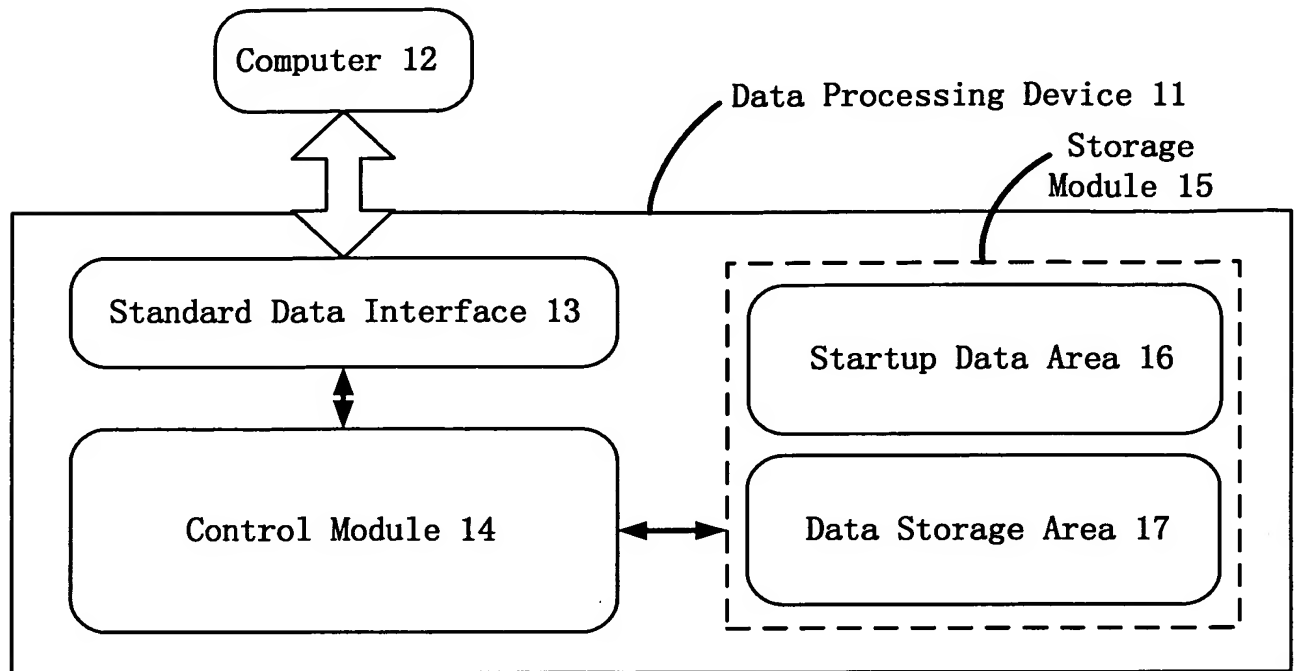


Fig.1

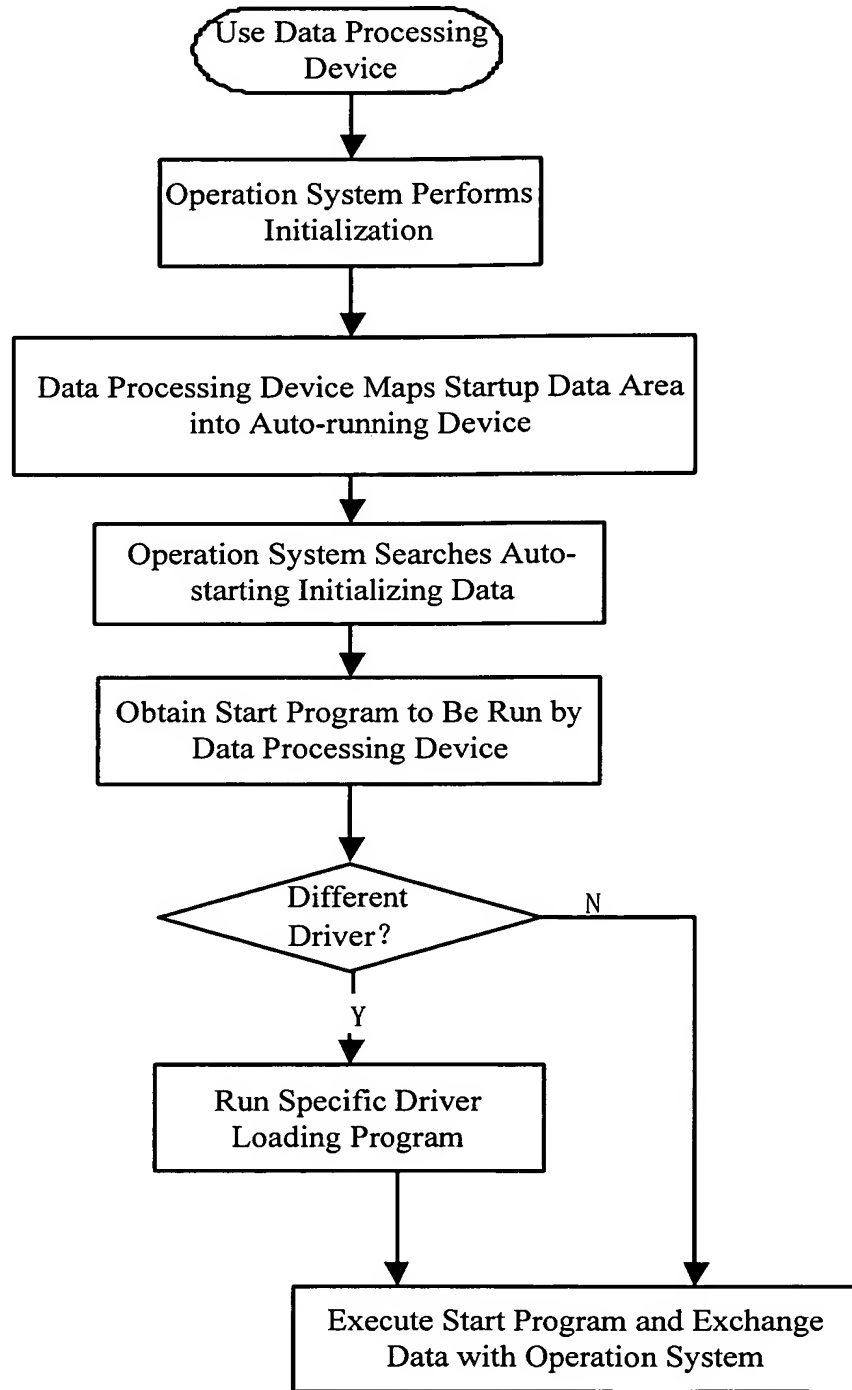


Fig.2

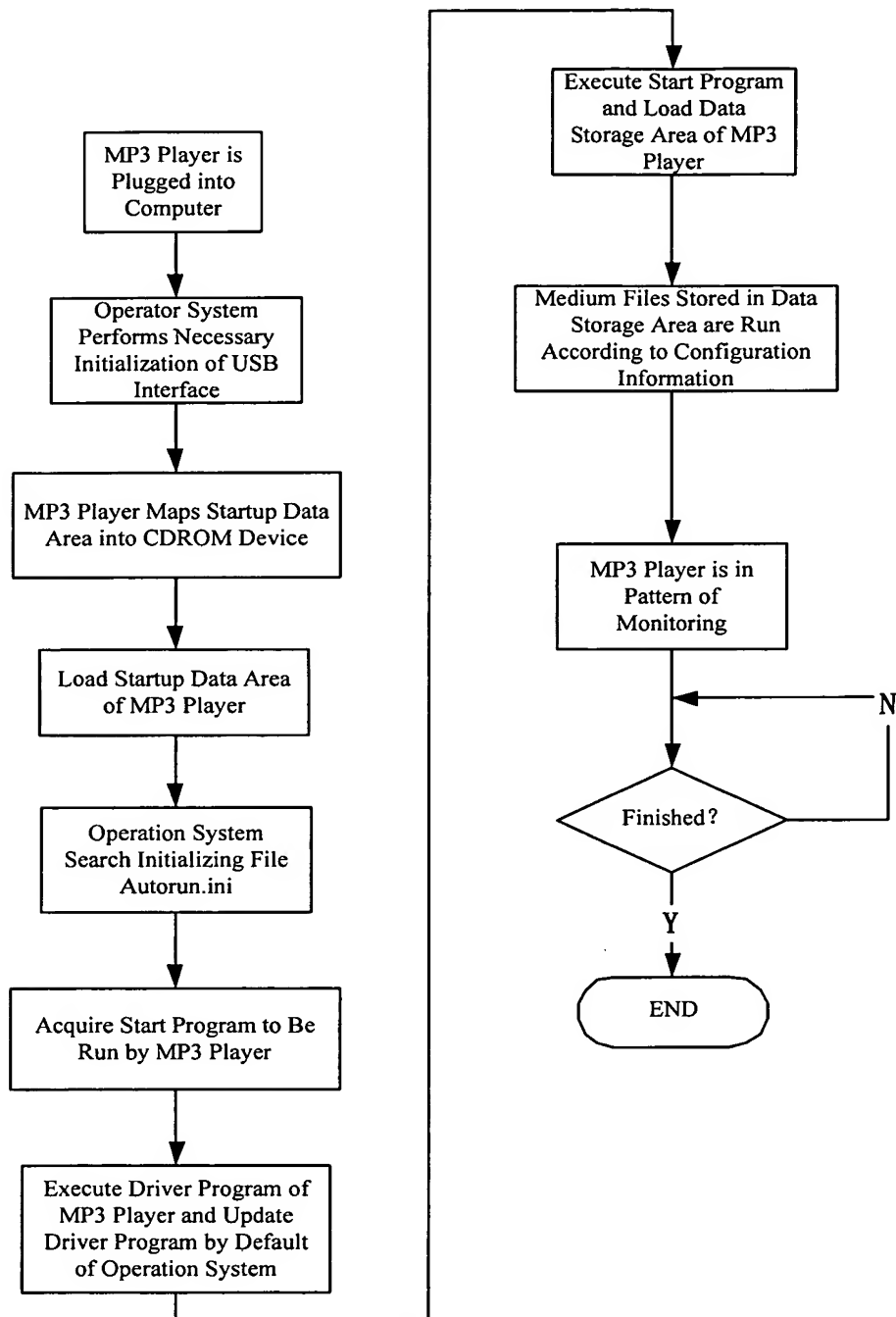


Fig.3

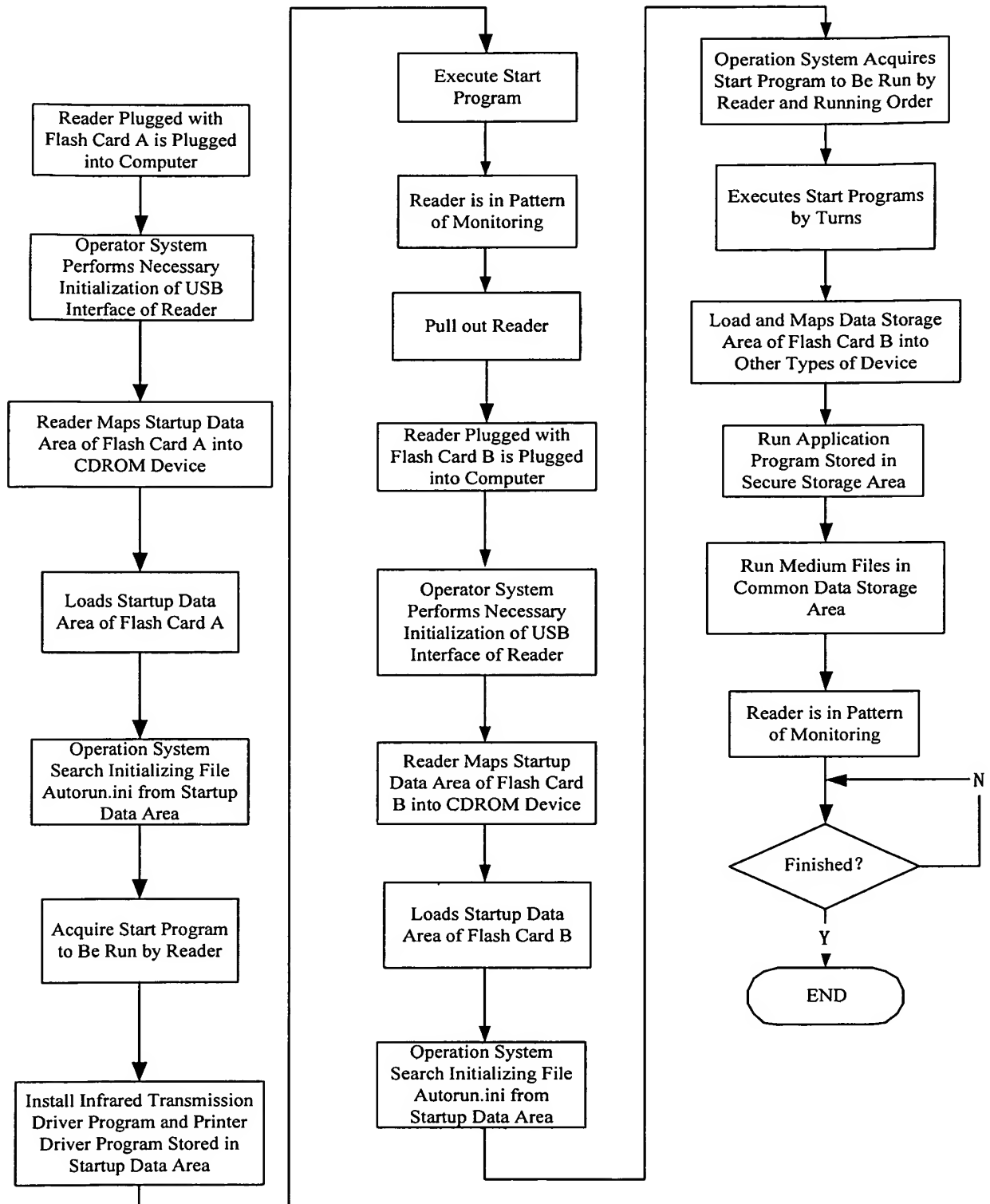


Fig.4

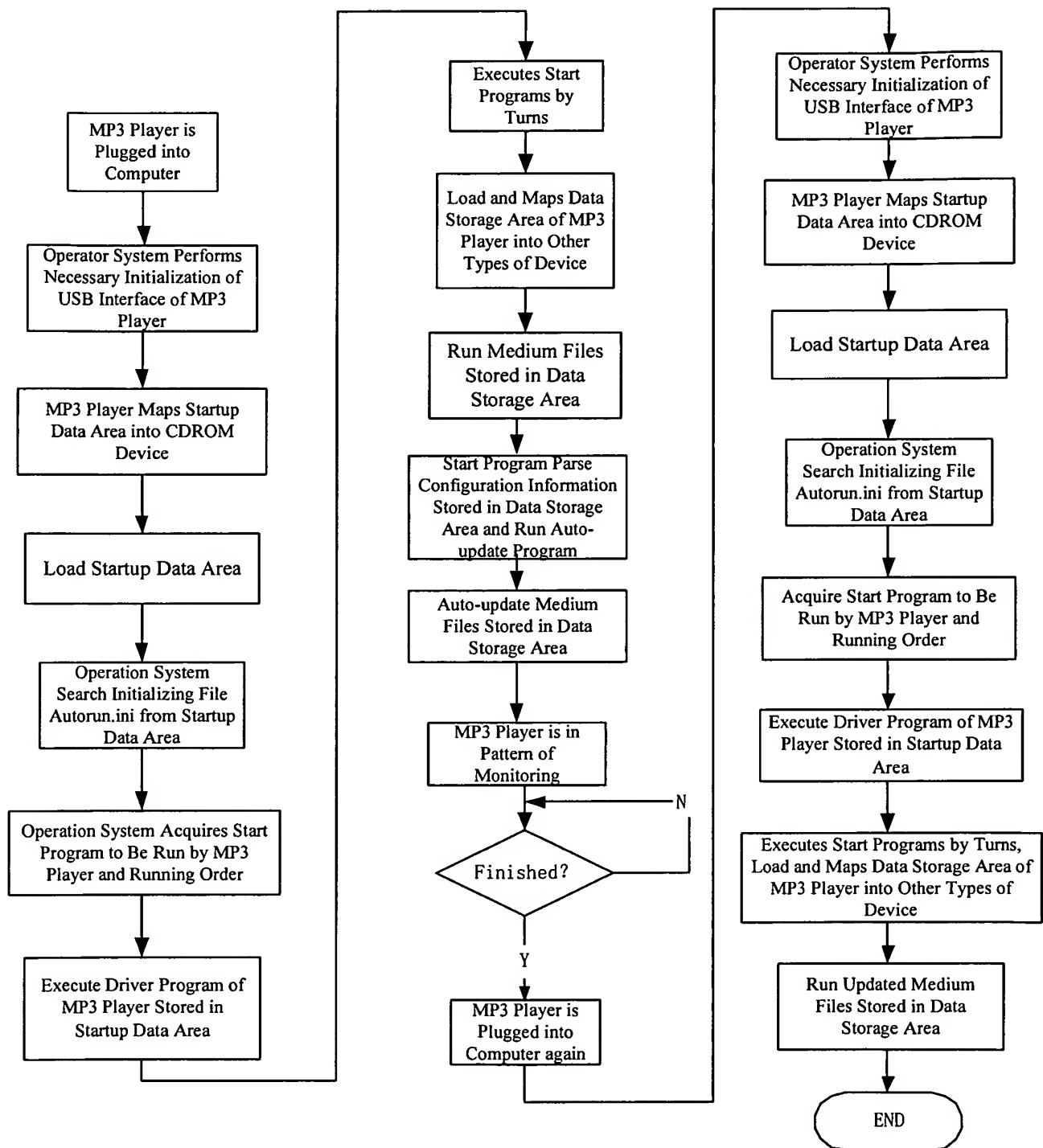


Fig.5